Use Cases

for

extraVEGANza

Version 1.3 approved

Prepared by Bryan Lu

Team Anyhow Anything Anywhere

9th March 2023

Revision History

Name	Date	Reason For Changes	Version
Bryan Lu	9/3/2023	First Version	1.1
Bryan Lu	18/3/2023	Refined some use cases	1.2
Bryan Lu	2/4/2023	Refined use cases	1.3

Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

1. Use Case Identification

1.1. Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1 Created By

Supply the name of the person who initially documented this use case.

1.3.2 Date Created

Enter the date on which the use case was initially documented.

1.3.3 Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.3.4 Date Last Updated

Enter the date on which the use case was most recently updated.

2. Use Case Definition

2.1. Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

2.2. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

2.3. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

2.4. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

2.5. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

2.6. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

2.7. Flow of Events

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

2.8. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by "AC" to indicate "Alternative Course". Example: X.Y.AC.1.

2.9. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by “EX” to indicate “Exception”. Example: X.Y.EX.1.

2.10. Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

2.11. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

2.12. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

2.13. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

3. Use Cases

3.1. Access Account

Use Case ID:	UC001		
Use Case Name:	Access Account		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-04-02

Actor:	User (Primary), User Database (Secondary)
Description:	This use case allows the user to access their account information, update the information, sign out or delete the account.
Preconditions:	1. The user must have already opened the application. 2. The application must have established a connection to the application's user database.
Postconditions:	1. The user’s account information must be shown in the account page and updated (optional). or 2. A login page must be shown. or 3. The user must be signed out of their account. or 4. The account must be deleted.
Priority:	1
Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the account button to open the account page. 2. The application retrieves the user information from the application’s user database. 3. The application displays the user information. 4. The user views and updates the information. 5. The user clicks on the save button to save the updated information. 6. The application updates the user’s information.
Alternative Flows:	AF001.1: At Step 1, if the user has not logged into an account.

	1. The application displays a login page. 2. The user login use case is implemented with the extended Log in / Sign up use case (UC105). AF001.2: At Step 4, if the user clicks on the sign out button. 1. The application signs the user out of the application. AF001.3: At Step 4, if the user clicks on the delete account button. 1. The application confirms the deletion by prompting the user to click on the button again. 2. The user clicks on the button again to confirm delete within 5 seconds, otherwise deletion process is cancelled. 3. The application requests the database to remove the user's account.
Exceptions:	N/A
Includes:	Log in / Sign up
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

3.2. Access Help

Use Case ID:	UC002		
Use Case Name:	Access Help		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-09

Actor:	User
Description:	This use case allows the user to access the help page of the application.
Preconditions:	1. The user must have already opened the application.
Postconditions:	1. The help page must be shown to the user.
Priority:	1

Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the help button to open the help page. 2. The application displays the help page to the user. 3. The user views the help page.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

3.3. Access Settings

Use Case ID:	UC003		
Use Case Name:	Access Settings		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-04-02

Actor:	User
Description:	This use case allows the user to change the settings for the application.
Preconditions:	1. The user must have already opened the application.
Postconditions:	1. The settings page must be shown to the user.
Priority:	1
Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the settings button to open the settings page. 2. The application retrieves the current settings from the local storage. 3. The application displays the user’s app settings. 4. The user views and updates the settings. 5. The application updates the settings accordingly.

Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	The application has only two settings, which is to reset map position, and to reset application.
Notes and Issues:	N/A

3.4. Access Map

Use Case ID:	UC004		
Use Case Name:	Access Map		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-09

Actor:	User (Primary)
Description:	This use case allows the user to access the map page of the application.
Preconditions:	1. The user must have already opened the application.
Postconditions:	1. The map page must be shown to the user.
Priority:	1
Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the map button to open the map page. 2. The application displays the map page to the user.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	Display Map
Special Requirements:	N/A
Assumptions:	N/A

Notes and Issues:	N/A

3.5. Access List

Use Case ID:	UC005		
Use Case Name:	Access List		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-18

Actor:	User (Primary)
Description:	This use case allows the user to access the list page of the application.
Preconditions:	1. The user must have already opened the application.
Postconditions:	1. The list page must be shown to the user.
Priority:	3
Frequency of Use:	1 to 3 times per use of application
Flow of Events:	1. The user clicks on the list button to open the list page. 2. The application displays the list page to the user. 3. The application retrieves the list of restaurants from the database. 4. The application displays the restaurants on the community page to the user.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

3.6. Access Community

Use Case ID:	UC006		
Use Case Name:	Access Community		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-18

Actor:	User (Primary), Application (Secondary)
Description:	This use case allows the user to access the community page of the application.
Preconditions:	1. The user must have already opened the application. 2. The application must have established a connection to the application's database.
Postconditions:	1. The community page and the community posts must be shown to the user.
Priority:	3
Frequency of Use:	1 to 3 times per use of application
Flow of Events:	1. The user clicks on the community button to open the list page. 2. The application displays the community page to the user. 3. The application retrieves the community posts from the database. 4. The application displays the community posts on the community page to the user.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

3.7. Select Dietary Restrictions

Use Case ID:	UC011		
Use Case Name:	Select Dietary Restrictions		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-04-02

Actor:	User
Description:	This use case allows the user to select their preferred dietary restrictions, which are Vegan, Vegetarian, Gluten-free, and Lactose-free.
Preconditions:	1. The user must have opened the map or list page.
Postconditions:	1. The application must have responded to the dietary restrictions that the user has chosen.
Priority:	2
Frequency of Use:	1 to 4 times per use of application
Flow of Events:	1. The user selects the dietary restrictions by checking the checkbox of the corresponding dietary restrictions choice. 2. The application updates the page according to the update restrictions choices.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	1. If none of the options are selected, the application assumes that all of the options are selected.
Notes and Issues:	N/A

3.8. Change Sort Order

Use Case ID:	UC012
Use Case Name:	Change Sort Order

Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-09

Actor:	User
Description:	This use case allows the user to change the sort order of the listings of restaurants in the list page, which includes the three order choices, ascending alphabetically (A - Z), descending alphabetically (Z - A), and rating (in descending order).
Preconditions:	1. The user must open the list page.
Postconditions:	1. The application must reorder the restaurant list according to the sort order that the user chose.
Priority:	2
Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the sorting button. 2. The application displays three choices in the dropdown menu that appears below the button. 3. The user selects their sorting choice. 4. The application reorders the restaurant listing accordingly.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	The initial sorting order would be ascending alphabetically (A - Z).
Notes and Issues:	N/A

3.9. Share Community Post

Use Case ID:	UC021		
Use Case Name:	Share Community Post		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-04-02

Actor:	User (Primary), Application Database (Secondary)
Description:	This use case allows the user to upload a post to the community page.
Preconditions:	1. The user must have opened the restaurant information page. 2. The application must have established a connection to the application's community database. 3. The user must have logged in to an account.
Postconditions:	1. The user must have successfully uploaded a community post. or 2. The application must have not receive the community post if the user is not logged in yet.
Priority:	2
Frequency of Use:	1 time per use of application
Flow of Events:	1. The user clicks on the share community post button. 2. The application displays an upload post page to the user. 3. The user fills in the details of the community post. 4. The user clicks on the submit button and submits the post to the application's database. 5. The application database records the user's post. 6. The application closes the upload post page.
Alternative Flows:	AF021.1: At Step 4, if the user did not fill in all necessary information. 1. The application prompts the user to input the necessary information. 2. Go to Step 3.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

3.10. Search Restaurants

Use Case ID:	UC022		
Use Case Name:	Search Restaurants		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-10

Actor:	User (Primary), Application Database (Secondary)
Description:	This use case allows the user to filter the list of restaurants that shown search for restaurants.
Preconditions:	1. The user must have already opened the list page. 2. The list page must have already retrieved the list of restaurants.
Postconditions:	1. The application must have filtered the restaurant list based on the search query.
Priority:	2
Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the search bar to search restaurants. 2. The user enters the search query into the search bar and submits it to the application. 3. The application uses the search query and filters the list of restaurants that contains the search query. 4. The application receives the filtered search results from the database.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	The application filters the restaurants based on whether the restaurant name contains the search query verbatim.
Notes and Issues:	N/A

3.11. Select Restaurant

Use Case ID:	UC023		
Use Case Name:	Select Restaurant		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-04-02

Actor:	User (Primary), Application Database (Secondary)
Description:	This use case allows the user to select and view the information of a restaurant, which includes the location, menu, ratings, etc..
Preconditions:	1. The user must be at the list page. 2. A list of restaurants on the list page, or a map of restaurants on the map page must be displayed to the user.
Postconditions:	1. The restaurant information must be displayed to the user.
Priority:	2
Frequency of Use:	1 to 5 times per use of application
Flow of Events:	1. The user clicks on the restaurant icon or clicks on the more info button of a restaurant. 2. The application retrieves the information of the restaurant from the application database. 3. The application displays the restaurant information to the user. 4. The user views the information displayed.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	If some information of the restaurant is missing, the application would simply not render out that particular information.
Notes and Issues:	N/A

3.12. Select Location

Use Case ID:	UC024		
Use Case Name:	Select Location		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-10

Actor:	User (Primary), Location Database (Secondary)
Description:	This use case allows the user to select a location to search for restaurants in that nearby area in the map page.
Preconditions:	1. The user must be in the map page. 2. The application must have established a connection to the application's database.
Postconditions:	1. The application must have received the location selected by the user.
Priority:	2
Frequency of Use:	1 to 3 times per use of application
Flow of Events:	1. The user enters a search query into the search bar. 2. The user presses “Enter”. 3. The application will display a dropdown menu of all possible search results. 4. The user clicks on one of the options. 5. The application receives the location that the user selected.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

3.13. Display Map

Use Case ID:	UC103		
Use Case Name:	Display Map		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-10

Actor:	Application, Google Map API
Description:	This use case allows the application to show a Google Map indicating the locations of restaurants to the user.
Preconditions:	1. The application must have received the user's location. 2. The application must have received a list of restaurants to be displayed on Google Map. 3. The application must have established a connection to the application's database, and Google Map API.
Postconditions:	1. A Google Map must be shown to the user indicating the locations of restaurants.
Priority:	3
Frequency of Use:	1 to 3 times per use of application
Flow of Events:	1. The application receives and confirms the list of restaurants to be displayed on Google Map, based on the location that the user entered. 2. The application retrieves the location information of the restaurants from its database. 3. The application sends the location data to Google Map API. 4. The application receives the data from Google Map API. 5. The application displays a Google Map with the information from the Google Map API to the user.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	Display Map
Special Requirements:	N/A
Assumptions:	If the user has not entered a location of their choice, the default location would be used instead.

Notes and Issues:	N/A

3.14. Search Community Posts

Use Case ID:	UC104		
Use Case Name:	Search Restaurants		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-04-02	Date Last Updated:	2023-04-02

Actor:	User (Primary), Application Database (Secondary)
Description:	This use case allows the user to filter the list of restaurants that shown search for restaurants.
Preconditions:	1. The user must have already opened the list page. 2. The list page must have already retrieved the list of restaurants.
Postconditions:	2. The application must have filtered the restaurant list based on the search query.
Priority:	2
Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the search bar to search community posts. 2. The user enters the search query into the search bar and submits it to the application. 3. The application uses the search query and filters the list of community posts that contains the search query. 4. The application receives the filtered search results from the database.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	The application filters the community posts based on whether the title, restaurant name, and content of the posts contains the search query verbatim.

Notes and Issues:	N/A

3.15. Log in / Sign up

Use Case ID:	UC024		
Use Case Name:	Log in / Sign up		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	2023-03-09	Date Last Updated:	2023-03-10

Actor:	User (Primary), Application (Secondary)
Description:	This use case allows the user to log in or sign up an account.
Preconditions:	1. The application must have established a connection to the application's database. 2. The user must be in the account page, and not logged in yet.
Postconditions:	1. The user must be logged in to their account.
Priority:	2
Frequency of Use:	1 time per use of application
Flow of Events:	1. The application shows the user an log in or sign up page. 2. The user enters the necessary information. 3. The user clicks on the login / sign up button. 4. The application authenticates the user. 5. The application logs the user in once authentication is successful. 6. The application closes the account page.
Alternative Flows:	AF024.1: At Step 3, if the user did not provide complete information. 1. The application prompts the user to complete the information. 2. Go to Step 2. AF024.2: At Step 4, if authentication failed. 1. The application will display an error message indicating the reason the authentication failed.

	2. Go to Step 2.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A